

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



MẠNG MÁY TÍNH (CO3093)

BÀI TẬP LỚN 1
DEVELOP A NETWORK APPLICATION

Giảng viên hướng dẫn: Hoàng Lê Hải Thanh
Nguyễn Phương Duy

Sinh viên thực hiện: Nguyễn Minh Linh - 2013630
Nguyễn Thái Bình - 1910843
Hà Trần Quang Trung - 2014874
Nguyễn Thành Chung - 2012741

Thành phố Hồ Chí Minh, Tháng 4 Năm 2024



Mục lục

1	Mô tả công việc	2
2	Phase 1: Cơ sở lý thuyết	3
2.1	Xác định và mô tả các chức năng của ứng dụng và các giao thức được sử dụng cho từng chức năng	3
2.1.1	Tìm kiếm và Tải xuống tệp	3
2.1.2	Tải lên tệp	3
2.1.3	Quản lý tệp và thư mục	4
2.1.4	Tạo tài khoản và Đăng nhập	5
2.1.5	Bảo mật và Quản lý người dùng	5
2.1.6	Giao diện người dùng thân thiện	6
3	Phase 2: Cơ sở lý thuyết	6
3.1	Các chức năng của ứng dụng	6
3.1.1	Server	6
3.1.2	Client	7
3.2	Các giao thức cho ứng dụng	8
3.2.1	Giữa các tính năng khi giao tiếp giữa client và server	8
3.2.2	Giữa các tính năng khi giao tiếp giữa client và client	8
3.3	Diagram	8
3.3.1	Use-case diagram	8
3.3.2	Class diagram	9
4	Kết quả hiện thực	10
4.1	Client1 dùng lệnh publish để chia sẻ thông tin file lên server	10
4.2	Client1 dùng lệnh fetch để lấy thông tin file từ server và tải từ các peers khác . .	11
4.3	Phía cơ sở dữ liệu	11
5	Kết quả đầu ra	12
6	Đánh giá hiệu năng	12
7	Kết luận	13
8	Nhật ký làm việc	14
9	Tài liệu tham khảo	15



1 Mô tả công việc

MEMBER	STUDENT ID	DESCRIPTION WORKS	EFFORT
Nguyễn Minh Linh	2013630	Thực hiện tìm hiểu về bài tập lớn, code Server, Client, thực hiện database, vẽ sơ đồ, làm báo cáo.	100%
Nguyễn Thái Bình	1910843	Thực hiện tìm hiểu về bài tập lớn.	20%
Hà Trần Quang Trung	2014874	Thực hiện tìm hiểu về bài tập lớn.	30%
Nguyễn Thành Chung	2012741	Thực hiện tìm hiểu về bài tập lớn, thực hiện làm báo cáo.	50%

2 Phase 1: Cơ sở lý thuyết

2.1 Xác định và mô tả các chức năng của ứng dụng và các giao thức được sử dụng cho từng chức năng

2.1.1 Tìm kiếm và Tải xuống tệp

- **Tìm kiếm tệp:** Người dùng sử dụng giao diện của ứng dụng để nhập các thông tin tìm kiếm như tên tệp, thể loại, hoặc từ khóa liên quan. Ứng dụng sau đó sẽ gửi yêu cầu tìm kiếm đến các nút khác trong mạng P2P hoặc đến máy chủ trung tâm (nếu có) để tìm kiếm các nguồn chia sẻ có chứa tệp tương ứng.
- **Phản hồi tìm kiếm:** Các nút hoặc máy chủ trung tâm nhận yêu cầu tìm kiếm và trả về kết quả tìm kiếm cho người dùng. Kết quả này có thể bao gồm danh sách các nguồn chia sẻ chứa tệp tìm kiếm, bao gồm địa chỉ IP, tên tệp, kích thước, và các thông tin khác.
- **Chọn nguồn chia sẻ:** Người dùng sau đó chọn một hoặc nhiều nguồn chia sẻ từ kết quả tìm kiếm để tải xuống tệp. Các nguồn này có thể là các máy tính khác trong mạng P2P hoặc các máy chủ trung tâm.
- **Kết nối và tải xuống:** Khi người dùng chọn một nguồn chia sẻ, ứng dụng sẽ tạo kết nối trực tiếp hoặc gián tiếp với nguồn đó. Sau đó, quá trình tải xuống bắt đầu, trong đó các phần nhỏ của tệp được yêu cầu và tải về từ nguồn chia sẻ.
- **Quản lý tải xuống:** Trong quá trình tải xuống, ứng dụng quản lý việc nhận các phần nhỏ của tệp từ nhiều nguồn khác nhau đồng thời để tối ưu hóa tốc độ tải xuống và đảm bảo tính đồng nhất của tệp.
- **Hoàn thành tải xuống:** Khi tất cả các phần của tệp đã được tải xuống, ứng dụng tự động kiểm tra và xác minh tính toàn vẹn của tệp, sau đó thông báo cho người dùng rằng quá trình tải xuống đã hoàn tất.

Giao thức truyền thông:

- **HTTP/HTTPS:** Sử dụng để tìm kiếm và tải xuống tệp thông qua các truy vấn HTTP.
- **BitTorrent:** Sử dụng cho các quy trình tìm kiếm và tải xuống dữ liệu trong mạng P2P.

2.1.2 Tải lên tệp

- **Chọn tệp cần chia sẻ:** Người dùng sử dụng giao diện của ứng dụng để chọn các tệp hoặc thư mục mà họ muốn chia sẻ với người dùng khác trên mạng. Đây có thể là các tệp đã có sẵn trên máy tính của họ.
- **Xác định phần trích dẫn (hash):** Trước khi tải tệp lên, ứng dụng sẽ tạo ra một phần trích dẫn hoặc hash cho các phần của tệp. Hash này được sử dụng để xác minh tính toàn vẹn của tệp trong quá trình tải xuống.
- **Phân tách tệp thành các phần nhỏ (piece):** Tệp được phân tách thành các phần nhỏ có kích thước cố định gọi là các piece. Mỗi piece được gắn với một mã hash để xác định nó.

- **Tạo tệp torrent:** Ứng dụng sẽ tạo một tệp metainfo (.torrent file) chứa thông tin về tệp, các phần trích dẫn (hash) của các piece, và các thông tin khác như tên tệp, kích thước, và các thông tin về máy chủ tracker.
- **Chia sẻ tệp torrent:** Sau khi tạo tệp torrent, người dùng có thể chia sẻ tệp này với người dùng khác thông qua email, trang web, hoặc các kênh khác trên mạng. Tệp torrent này chứa thông tin cần thiết để người dùng khác có thể tìm kiếm và tải tệp từ nguồn chia sẻ.
- **Kết nối và chia sẻ thông tin:** Khi người dùng khác tìm thấy tệp torrent và muốn tải tệp xuống, họ sẽ kết nối với máy tính của người chia sẻ thông qua mạng BitTorrent. Ứng dụng sẽ gửi các phần nhỏ của tệp (pieces) cho người tải xuống từ máy tính của mình.

Giao thức truyền thông:

- **HTTP/HTTPS:** Sử dụng để tải lên tệp lên máy chủ thông qua các yêu cầu HTTP.
- **BitTorrent:** Cung cấp khả năng chia sẻ tệp trực tiếp với người dùng khác trong mạng P2P.

2.1.3 Quản lý tệp và thư mục

- **Tạo và quản lý thư mục:** Người dùng có thể tạo và quản lý các thư mục trên máy tính của họ để tổ chức tệp tin. Họ có thể di chuyển, đổi tên, sao chép, và xóa các thư mục này theo nhu cầu.
- **Tạo và quản lý tệp:** Người dùng có thể tạo và quản lý các tệp tin trong các thư mục đã tạo. Họ có thể tải lên, tải xuống, đổi tên, sao chép, và xóa các tệp tin. Họ cũng có thể xem thông tin chi tiết về mỗi tệp như kích thước, ngày tạo, ngày sửa đổi, và định dạng tệp.
- **Tải tệp từ các nguồn khác nhau:** Người dùng có thể tải tệp từ các nguồn khác nhau trên mạng BitTorrent. Họ có thể tìm kiếm tệp thông qua các trang web torrent hoặc qua tính năng tìm kiếm được tích hợp trong ứng dụng của mình.
- **Quản lý tải xuống:** Người dùng có thể quản lý các tệp đang được tải xuống. Họ có thể kiểm tra tiến độ tải xuống, tạm dừng, tiếp tục, hoặc hủy bỏ các tải xuống đang diễn ra.
- **Quản lý tải lên:** Người dùng có thể quản lý các tệp đang được tải lên từ máy tính của họ. Họ có thể kiểm tra tiến độ tải lên, tạm dừng, tiếp tục, hoặc hủy bỏ các tải lên đang diễn ra.
- **Tích hợp với các tính năng định vị:** Một số ứng dụng BitTorrent cung cấp tính năng định vị để cho phép người dùng theo dõi các tệp và thư mục đã chia sẻ hoặc tải xuống từ các máy tính khác trên mạng. Điều này giúp họ dễ dàng quản lý và truy cập vào các tệp đã chia sẻ trên nhiều thiết bị.

Giao thức truyền thông:

- **HTTP/HTTPS:** Sử dụng để thực hiện các thao tác quản lý tệp và thư mục thông qua các yêu cầu HTTP.
- **BitTorrent:** Cung cấp khả năng quản lý tệp và thư mục trên máy chủ và trong mạng P2P.

2.1.4 Tạo tài khoản và Đăng nhập

- **Tạo tài khoản:** Người dùng có thể tạo một tài khoản cá nhân trên ứng dụng để có quyền truy cập vào các tính năng và dịch vụ cao cấp. Quá trình tạo tài khoản thường bao gồm nhập các thông tin cá nhân như tên, địa chỉ email, và mật khẩu.
- **Xác thực tài khoản:** Sau khi tạo tài khoản, người dùng sẽ nhận được một email xác thực để kích hoạt tài khoản. Họ cần nhấp vào liên kết xác thực trong email để hoàn tất quá trình đăng ký.
- **Đăng nhập:** Người dùng có thể đăng nhập vào tài khoản của mình bằng cách nhập địa chỉ email và mật khẩu đã đăng ký trong quá trình tạo tài khoản. Sau khi đăng nhập thành công, họ sẽ có quyền truy cập vào các tính năng và dịch vụ mà ứng dụng cung cấp.
- **Quên mật khẩu:** Nếu người dùng quên mật khẩu, họ có thể sử dụng tính năng "Quên mật khẩu" để đặt lại mật khẩu. Họ cần nhập địa chỉ email được liên kết với tài khoản của họ, sau đó hệ thống sẽ gửi một liên kết đặt lại mật khẩu đến địa chỉ email đó.
- **Bảo mật và quản lý tài khoản:** Người dùng có thể quản lý thông tin cá nhân và bảo mật của tài khoản của mình. Họ có thể thay đổi mật khẩu, cập nhật thông tin cá nhân, và quản lý các thiết lập bảo mật như xác thực hai yếu tố (2FA).

Giao thức truyền thông:

- **HTTPS:** Sử dụng để bảo vệ quá trình tạo tài khoản và đăng nhập thông qua kết nối an toàn.
- **OAuth:** Sử dụng để cung cấp quy trình xác thực dựa trên token giữa các ứng dụng khác nhau.

2.1.5 Bảo mật và Quản lý người dùng

- **Quản lý tài khoản:** Người dùng có thể đăng ký và đăng nhập vào tài khoản của họ để truy cập vào các tính năng bảo mật và quản lý cá nhân. Họ có thể thay đổi mật khẩu, cập nhật thông tin cá nhân, và quản lý các thiết lập bảo mật như xác thực hai yếu tố (2FA).
- **Xác thực người dùng:** Ứng dụng cung cấp các phương pháp xác thực người dùng như địa chỉ email xác thực hoặc xác thực hai yếu tố (2FA) để bảo vệ tài khoản khỏi việc truy cập trái phép.
- **Quản lý quyền truy cập:** Quản trị viên có thể quản lý quyền truy cập của người dùng trong hệ thống. Họ có thể gán quyền truy cập cho các nhóm người dùng hoặc cho từng người dùng cụ thể, điều chỉnh quyền truy cập vào các tính năng và dịch vụ khác nhau.
- **Phân quyền:** Hệ thống phân quyền cho phép quản trị viên cấp quyền truy cập và quản lý các quyền hạn của người dùng trong ứng dụng. Các quyền hạn có thể bao gồm quyền tải lên, quyền tải xuống, quyền chỉnh sửa, và quyền xóa tệp.
- **Bảo mật dữ liệu:** Dữ liệu tải lên và tải xuống được mã hóa để bảo vệ khỏi việc truy cập trái phép. Các biện pháp bảo mật như mã hóa dữ liệu và SSL/TLS được triển khai để đảm bảo an toàn cho thông tin cá nhân của người dùng.

Giao thức truyền thông:

- HTTPS: Sử dụng để bảo vệ thông tin người dùng và dữ liệu thông qua kết nối an toàn.
- JSON Web Tokens (JWT): Sử dụng để xác thực và ủy quyền người dùng trong các yêu cầu HTTP.

2.1.6 Giao diện người dùng thân thiện

- Thiết kế giao diện đơn giản: Giao diện người dùng được thiết kế để dễ sử dụng và dễ hiểu, với các nút và menu được sắp xếp một cách logic và trực quan.
- Tích hợp hướng dẫn sử dụng: Giao diện cung cấp các hướng dẫn và gợi ý để người dùng có thể sử dụng ứng dụng một cách hiệu quả. Các hướng dẫn này có thể bao gồm các hộp thoại thông báo, video hướng dẫn, hoặc tài liệu trực tuyến.
- Tùy chỉnh giao diện: Người dùng có thể tùy chỉnh giao diện theo sở thích cá nhân, bao gồm chủ đề màu sắc, cỡ chữ, kiểu font, và cấu trúc cửa sổ.
- Hỗ trợ đa ngôn ngữ: Giao diện hỗ trợ nhiều ngôn ngữ để phục vụ người dùng trên toàn thế giới. Người dùng có thể chọn ngôn ngữ ưa thích của họ từ danh sách các ngôn ngữ được hỗ trợ.
- Tích hợp phản hồi người dùng: Giao diện cung cấp các cơ chế để người dùng có thể gửi phản hồi, báo cáo lỗi, hoặc đề xuất tính năng mới. Điều này giúp cải thiện trải nghiệm người dùng và phát triển ứng dụng theo hướng tích cực.

Giao thức truyền thông:

- HTML/CSS/JavaScript: Sử dụng để phát triển giao diện người dùng trên trình duyệt web.
- WebSocket: Cung cấp khả năng tương tác thời gian thực giữa trình duyệt và máy chủ.

3 Phase 2: Cơ sở lý thuyết

3.1 Các chức năng của ứng dụng

3.1.1 Server

Chức năng quản lý:

Máy chủ trở thành Quản lý của hệ thống truyền file, nhằm theo dõi client đang kết nối, file mà client đang giữ.

Lưu trữ danh sách các client đang kết nối với server và danh sách file của các client đấy:

Tính năng nhằm quản lý các file nằm ở máy hostname. Server tạo danh sách các client đã kết nối và các file của client đã kết nối. Khi client kết nối với server, server có thể theo dõi thông tin như địa chỉ IP, Port, trạng thái của client. Thông tin này giúp server xác định và quản lý các client một cách hiệu quả nhằm tránh, ngăn chặn các cuộc tấn công mạng hoặc giả mạo IP Address.

Ngoài ra server còn tạo một danh sách để quản lý và tìm kiếm các file mà client đã đưa lên thông qua lệnh publish. Danh sách này có thể bao gồm thông tin của file như tên file, kích thước file, thông tin các piece như kích thước, mã hash của từng phần trong file, tên máy client chứa file hostname, địa chỉ IP và port đang chứa file. Bằng cách theo dõi thông tin này, server có thể dễ dàng biết và trả lời được thông tin về máy đang giữ file mà các máy khác yêu cầu đang cần.

Trả lời yêu cầu của các client:

Server và các client được kiến trúc theo mô hình Server-Client:

- Client gửi yêu cầu: Yêu cầu này là yêu cầu fetch hoặc yêu cầu publish.
- Server nhận yêu cầu: Server xử lý yêu cầu đến. Nó quyết định những việc cần làm dựa trên loại yêu cầu, dữ liệu được cung cấp, thông tin xác thực của client.
- Server xử lý yêu cầu: Server thực hiện các yêu cầu mà client yêu cầu xử lý phù hợp.
- Server tạo và gửi phản hồi: Sau khi xử lý yêu cầu, Server sẽ tổng hợp mọi thứ cần thiết cho phản hồi. Thông tin cụ thể sẽ tùy thuộc vào yêu cầu nhưng thông tin này thường bao gồm siêu dữ liệu có thể có và dữ liệu. Sau đó, server sẽ gửi phản hồi này lại cho client bằng giao thức thích hợp.
- Client nhận phản hồi: Client nhận phản hồi của server, lưu file đã tải xuống, v.v

3.1.2 Client

Thông báo cho server biết những file đang có

Tính năng này sẽ gửi thông tin những file đang có ở máy Client lên server, sẽ có 2 trường hợp xảy ra:

- Nếu kết nối được đến Server, Client sẽ gửi danh sách tên file lên Server.
- Nếu không, Client sẽ báo lỗi.

Gửi thông tin để truy hồi file

Tính năng này sẽ gửi thông tin (tên file) lên Server, và sau đó Server sẽ thực hiện tìm kiếm file này ở các máy Client đang kết nối với Server và ở cơ sở dữ liệu. Sẽ có 2 trường hợp xảy ra:

- Nếu tìm ra, Server sẽ gửi về Client đang thực hiện yêu cầu các thông tin về file (tên của file ở local, file đang ở client nào). Sau đó, nếu người dùng ở máy Client đang thực hiện yêu cầu muốn tải về file đấy, file sẽ được chuyển thẳng giữa 2 máy Client, không thông qua Server.
- Nếu không tìm ra, hệ thống sẽ báo lỗi và kết thúc quá trình tìm kiếm. Khi này, nếu người dùng muốn, có thể nhập lại tên file và nhấn nút để tiếp tục thực hiện tìm kiếm mới.

Trình thông dịch dòng lệnh: "publish", "fetch"

Hệ thống sẽ có một phần hỗ trợ 2 lệnh là publish và fetch cụ thể như sau:

- publish file_name: Client có file tên file_name sẽ được thêm vào cơ sở dữ liệu của server với nhiều thông tin liên quan đến client và file đó.
- fetch file_name: Gửi yêu cầu tìm kiếm file file_name lên server, nếu tìm thấy thì client sẽ nhận được thông tin kết nối của client đang chứa tập tin đấy và tiến hành truyền tải dữ liệu.

3.2 Các giao thức cho ứng dụng

3.2.1 Giữa các tính năng khi giao tiếp giữa client và server

Các tính năng có sự giao tiếp giữa client và server sẽ sử dụng chung giao thức kết nối là TCP vì các thao tác giữa clients server có thể chấp nhận độ trễ nhất định tuy nhiên yêu cầu ở server như là một trung gian nhằm quản lý cơ sở dữ liệu và kết nối giữa các client với nhau nên cần đảm bảo thông tin chính xác và tránh bị mất gói.

Publish

Khi một Client sử dụng lệnh publish, Client sẽ gửi một yêu cầu tới Server muốn lưu trữ một file tên file này trên repository của mình là file_name. Sau đó từ Server sẽ tiến hành cập nhật cơ sở dữ liệu và gửi một xác nhận về lại cho Client.

Fetch

Khi một Client sử dụng lệnh publish, Client sẽ gửi một yêu cầu tới Server muốn truy xuất một file có tên file_name về máy mình. Server sẽ tiến hành truy vấn cơ sở dữ liệu để tìm ra các Client khác hiện đã publish file file_name lên repository của mình và tiến hành gửi lại cho Client đã gửi yêu cầu để Client đó tự tạo kết nối với Client đang giữ file để tiến hành chia sẻ file.

3.2.2 Giữa các tính năng khi giao tiếp giữa client và client

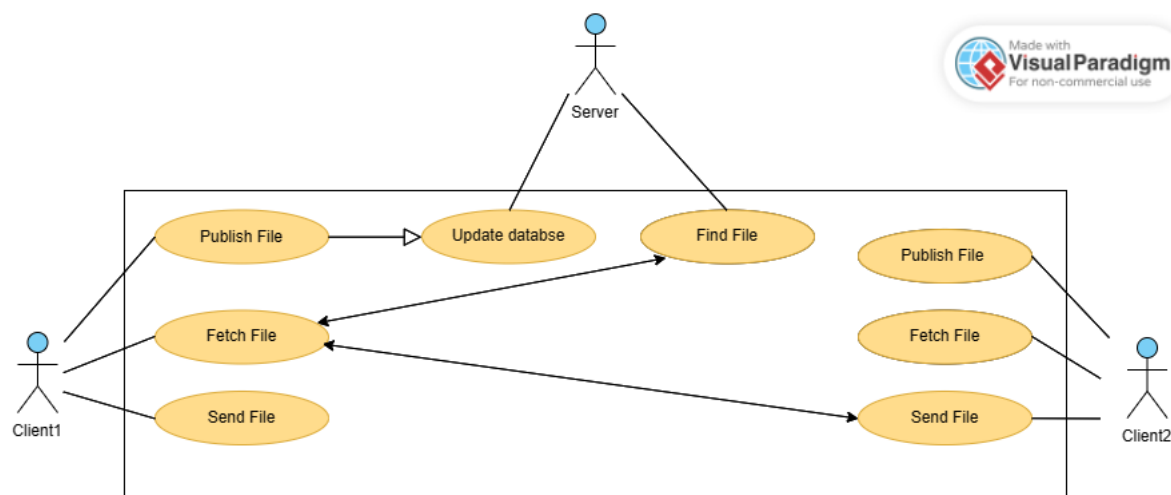
Các tính năng có sự giao tiếp giữa client và client sẽ sử dụng chung giao thức kết nối là TCP vì các thao tác giữa clients trong các chức năng này có thể chấp nhận độ trễ nhất định tuy nhiên yêu cầu việc dữ liệu khi nhận về cần đầy đủ, không bị mất gói.

Fetch

Khi Client đã kết nối đến peer, tức là một Client khác, để gửi yêu cầu được chia sẻ file file_name trên máy của peer. Peer sẽ tiến hành gửi lại nội dung trong file được yêu cầu về cho Client đã kết nối.

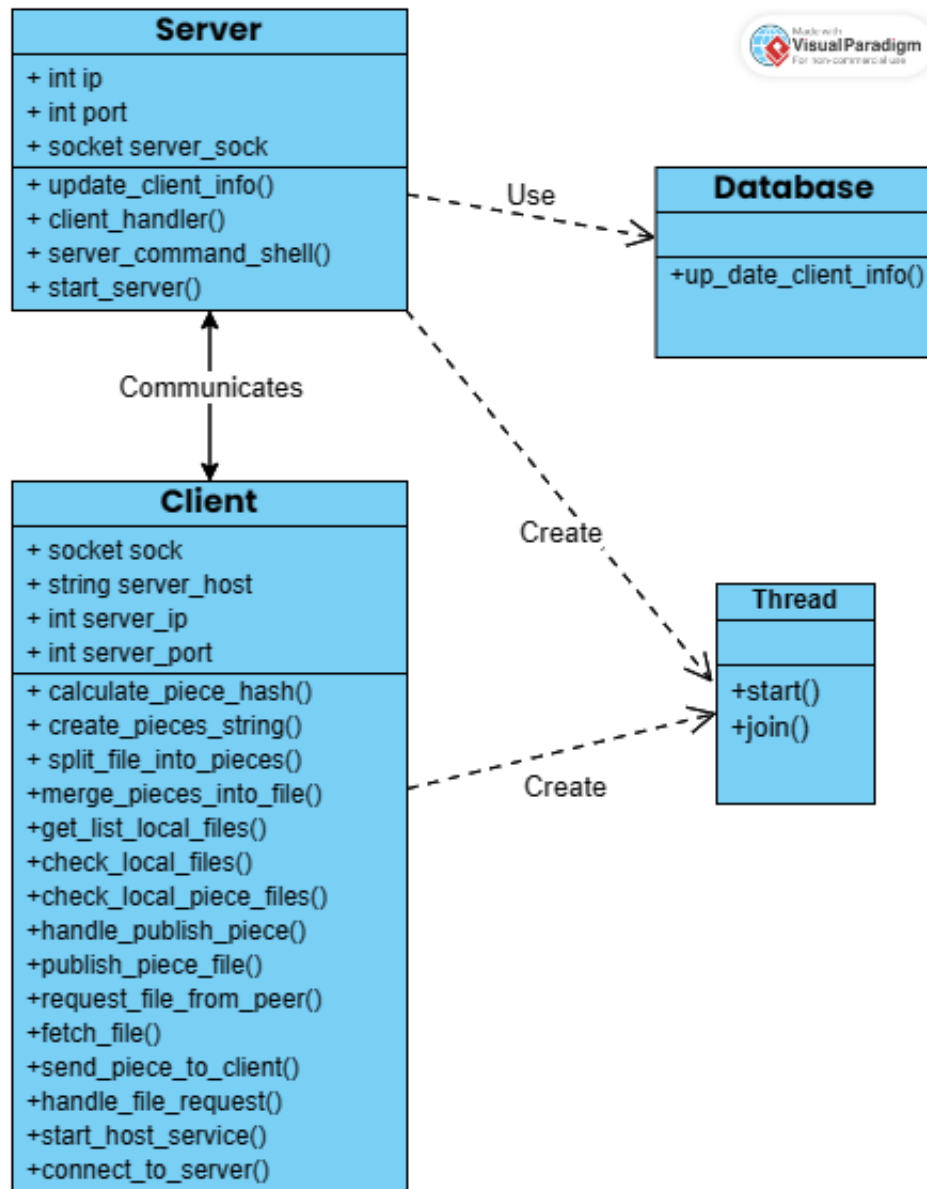
3.3 Diagram

3.3.1 Use-case diagram



Hình 1: Use-case diagram tổng

3.3.2 Class diagram



Hình 2: Class diagram tổng

4 Kết quả hiện thực

4.1 Client1 dùng lệnh publish để chia sẻ thông tin file lên server

Example

Client1 publish Data.pdf on Server

Server side

```
\server> python server.py
2024-04-25 15:51:31,649 - INFO - Server started and is listening for connections.
Server command: 2024-04-25 15:51:35,938 - INFO - Active connections: 2
2024-04-25 15:51:35,938 - INFO - Connection established with linh/192.168.56.1:65433)
2024-04-25 15:51:41,464 - INFO - Active connections: 3
2024-04-25 15:51:41,472 - INFO - Connection established with linh/192.168.56.1:65434)
2024-04-25 15:52:03,987 - INFO - Updating client info in database for hostname: linh/192.168.56.1:65433
2024-04-25 15:52:03,992 - INFO - Database update complete for hostname: linh/192.168.56.1:65433
```

Client side

```
\client1> python client.py
Enter command (publish file_name/ fetch file_name/ exit): publish Data.pdf
File Data.pdf have ['Data.pdf_piece1', 'Data.pdf_piece2', 'Data.pdf_piece3', 'Data.pdf_piece4', 'Data.pdf_piece5']
piece: ["b'\xe1\xd8\\r\\xa2\\t\\xee\\x0c\\xfd91\\x12\\xedkk\\xf1u\\x03\\x9b\\x14'",
'b"f\\xc3\\xc108\\xd7\\xb0ft\\xc5\\xf1\\x1e`7-\\'\\xa8!\\xf1\\x96"',
"b'cT\\xdf\\xc3~\\xf1d0\\x135\\x1d\\x0eu\\x95\\xd3\\xfb\\xd9*\\n\\x9a'",
'b'\\x0f\\xda\\xb7\\xda\\xda3\\x90Rr\\x88\\xc4\\xc0\\x85\\xfe]\\x0fN\\xf98\\'',
"b'\\tA\\xb9I\\x17\\xa0\\xa7\\xcF\\xeal\\xa1!\\x97\\xd6C-44\\xa7\\xd9\\xb6'"].
Please select num piece in file to publish:1 3 4
You was selected:
Number 1 : b'\xe1\xd8\\r\\xa2\\t\\xee\\x0c\\xfd91\\x12\\xedkk\\xf1u\\x03\\x9b\\x14'
Number 3 : b'cT\\xdf\\xc3~\\xf1d0\\x135\\x1d\\x0eu\\x95\\xd3\\xfb\\xd9*\\n\\x9a'
Number 4 : b'\\x0f\\xda\\xb7\\xda\\xda3\\x90Rr\\x88\\xc4\\xc0\\x85\\xfe]\\x0fN\\xf98'
File list updated successfully.
Enter command (publish file_name/ fetch file_name/ exit):
```

Hình 3: Ví dụ lệnh publish

Quá trình thực hiện như sau:

- Đầu tiên sau khi khởi động server và client1 gửi thông tin kết nối đến server.
- Sau đó client1 dùng lệnh publish để chia sẻ thông tin file Data.pdf.
- Client1 sẽ chia file Data.pdf ra các pieces, mỗi piece kích thước 512kb. Và yêu cầu người dùng chọn chia sẻ những phần nào.
- Người dùng chọn chia sẻ phần 1,3,4 trong 5 phần.
- Server thông báo Database đã được cập nhật.

4.2 Client1 dùng lệnh fetch để lấy thông tin file từ server và tải từ các peers khác

```
Client2 fetch Data.pdf

Server side

\server> python server.py
2024-04-25 15:51:31,649 - INFO - Server started and is listening for connections.
Server command: 2024-04-25 15:51:35,938 - INFO - Active connections: 2
2024-04-25 15:51:35,938 - INFO - Connection established with linh/192.168.56.1:65433)
2024-04-25 15:51:41,464 - INFO - Active connections: 3
2024-04-25 15:51:41,472 - INFO - Connection established with linh/192.168.56.1:65434)
2024-04-25 15:52:03,987 - INFO - Updating client info in database for hostname: linh/192.168.56.1:65433
2024-04-25 15:52:03,992 - INFO - Database update complete for hostname: linh/192.168.56.1:65433

Client side

client2> python client.py
Enter command (publish file_name/ fetch file_name/ exit):fetch Data.pdf
Hosts with the file Data.pdf:
Number: 1 linh/192.168.56.1:65433 piece_hash: b'\xe1\xd8\\r\xa2\t\xee\x0c\xfd91\x12\xedkk\xf1u\x03\x9b\x14' file_size:
2296627 piece_size: 524288 num_order_in_file: 1
Number: 3 linh/192.168.56.1:65433 piece_hash: b'cT\xdf\xc3~\xf1d0\x135\x1d\x0eu\x95\xd3\xfb\xd9*\n\x9a' file_size: 2296627
piece_size: 524288 num_order_in_file: 3
Number: 4 linh/192.168.56.1:65433 piece_hash: b'\x0f\xda\xb7\xda\xda3\x90Rr\x88\xc4\xc0\x85"\xfe]\x0fN\xf98' file_size:
2296627 piece_size: 524288 num_order_in_file: 4
```

Hình 4: Ví dụ lệnh fetch

Quá trình thực hiện như sau:

- Đầu tiên sau khi khởi động server và client2 gửi thông tin kết nối đến server.
- Sau đó client2 dùng lệnh fetch để lấy thông tin file Data.pdf và tải các phần xuống.
- Client2 được server phản hồi về thông tin có 3 phần của file Data.pdf gồm phần 1,3,4 đang được chia sẻ và chưa có trên Client2.
- Người dùng nhập các phần muốn tải vào. Sau đó Client2 kết nối tự động đến Client1 để tải các phần được chọn.
- Sau khi tải xuống ta có 3 phần gồm: Data.pdf_piece1, Data.pdf_piece3, Data.pdf_piece4. Tuy nhiên chưa đủ tất cả các phần để tạo thành file Data.pdf nên Client2 chưa tạo.
- Chúng ta có thể đợi Client1 hoặc các Client khác chia sẻ các phần khác để có đầy đủ. Khi tải xuống đầy đủ thì Client2 tự động ghép các file lại thành 1 file gốc.

4.3 Phía cơ sở dữ liệu

Tạo cơ sở dữ liệu và schema peers có bảng để lưu giữ các thông tin liên quan đến tập tin (peers_ip, peers_port, peers_hostname, file_name, file_size, piece_hash, piece_size,



num_order_in_file) thực hiện kết nối đến cơ sở dữ liệu. Hệ quản lý cơ sở dữ liệu mà nhóm sử dụng là PostgreSQL.

```
postgres=# SELECT * FROM peers;
```

peers_ip	peers_port	peers_hostname	file_name	file_size	piece_hash
192.168.56.1	65433	linh	Data.pdf	2296627	b'\xe1\xd8\r\xa2\t\xee\x0c\xfd91\x12\xedkk\xfu\x03\x9b\x14'
192.168.56.1	65433	linh	Data.pdf	2296627	b'cT\xdf\xc3~\xf1d0\x135\x1d\x0eu\x95\xd3\xfb\x9*\n\x9a'
192.168.56.1	65433	linh	Data.pdf	2296627	b'\x0f\xda\xb7\xda\xda3\x90Rr\x88\xc4\xc0\x85"\xfe]\x0fN\x98'

(3 rows)

Hình 5: Cấu trúc về cơ sở dữ liệu

5 Kết quả đầu ra

Ứng dụng chạy được trên hệ điều hành máy tính phổ biến là Window.

Về phía Server:

- Thực hiện được việc truy vấn, tương tác với cơ sở dữ liệu.
- Lưu giữ được thông tin của các client để tiện cho việc xác thực thông tin.
- Thực hiện được các thao tác truyền, gửi, nhận dữ liệu.

Về phía Client:

- Thực hiện được việc truyền, nhận dữ liệu từ client lên máy chủ và ngược lại.
- Thực hiện được việc truyền, nhận dữ liệu giữa các client với nhau.
- Thực hiện được các lệnh fetch, publish.

6 Đánh giá hiệu năng

Tất cả các chương trình ở điều hành Window đều hoạt động tốt.

Về phía server, tương tác giữa server và database rất nhanh và chính xác, thao tác tìm kiếm và trả kết quả phản hồi về phía Client rất nhanh.

Về phía client, lệnh publish được hiện thực rất nhanh, tuy nhiên lệnh fetch sẽ rất phụ thuộc vào tình trạng mạng và kích thước file. Quá trình tách file và merge file tương đối nhanh và không xảy ra bị lỗi trong quá trình.

Về mặt truy vấn cơ sở dữ liệu, các thao tác CRUD trên cơ sở dữ liệu có tốc độ thao tác nhanh và không xảy ra bất cứ tình trạng chậm trễ nào.

7 Kết luận

Nhóm chúng tôi đã hoàn thành dự án ứng dụng truyền tải dữ liệu peer-to-peer sử dụng ngôn ngữ lập trình Python. Dự án này mang đến cho chúng tôi cơ hội trau dồi kỹ năng lập trình, quản lý dữ liệu.

Thông qua quá trình thực hiện dự án, nhóm đã học được cách lập trình socket bằng Python. Chúng tôi đã thành thạo các kỹ thuật kết nối, truyền tải và nhận dữ liệu giữa các máy tính, vận dụng linh hoạt vào việc xây dựng ứng dụng truyền tải tập tin.

Bên cạnh đó, nhóm còn nắm vững kỹ năng quản lý dữ liệu. Chúng tôi đã học cách truy vấn, lưu trữ thông tin truyền và nhận dữ liệu vào cơ sở dữ liệu một cách khoa học và hiệu quả, đảm bảo tính chính xác và bảo mật cho thông tin.

Nhìn về tương lai, nhóm dự định tiếp tục phát triển ứng dụng, tập trung vào các khía cạnh sau:

- Nâng cao tính bảo mật: Mã hóa dữ liệu khi truyền và nhận tập tin, đồng thời mã hóa dữ liệu nhạy cảm của người dùng khi lưu trữ trong cơ sở dữ liệu để đảm bảo an toàn thông tin.
- Tăng cường chức năng: Thêm các tính năng mới như quản lý danh sách bạn bè, chia sẻ tập tin theo nhóm, v.v. để đáp ứng nhu cầu đa dạng của người dùng.
- Cải thiện hiệu suất: Tối ưu hóa hiệu suất truyền tải dữ liệu và sử dụng tài nguyên hệ thống hiệu quả hơn để mang đến trải nghiệm tốt nhất cho người dùng.

Dự án ứng dụng truyền tải dữ liệu peer-to-peer là một trải nghiệm học tập và rèn luyện quý giá đối với nhóm. Dự án đã giúp chúng tôi phát triển kỹ năng lập trình, quản lý dữ liệu.



8 Nhật ký làm việc

- Ngày **12/04/2024**: Họp thảo luận chia nhiệm vụ của bài tập lớn cho các thành viên, tạo báo cáo trên Latex.
- Ngày **13/04/2024 - 23/04/2024**: Thực hiện nghiên cứu bài tập lớn, thực hiện code.
- Ngày **24/04/2024 - 25/04/2024**: Thực hiện code, thực hiện làm báo cáo.



9 Tài liệu tham khảo

- [1] BitTorrentSpecification. Được truy cập từ: <https://wiki.theory.org/BitTorrentSpecification>
- [2] Tracker scrape. Được truy cập từ: https://en.wikipedia.org/wiki/Tracker_scrape